Bay-Delta SELFE-W

Workflow Guide

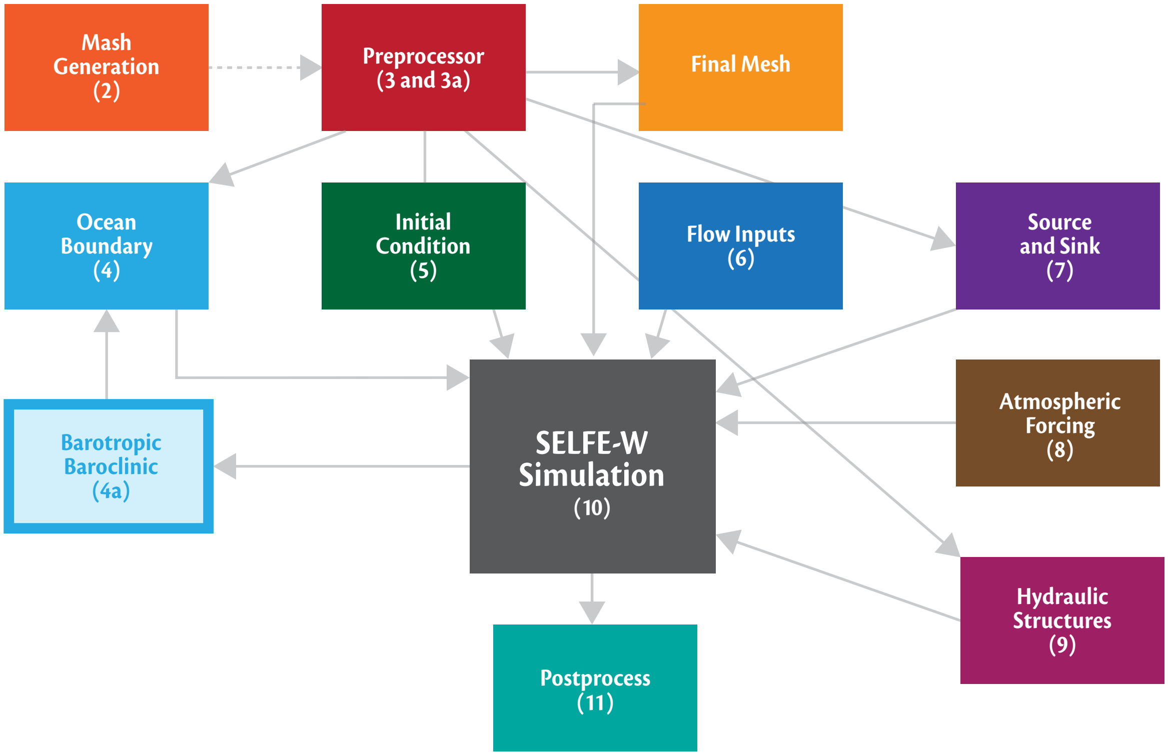
This workflow guide provides an overview of the Bay-Delta SELFE modeling process and resources. It is targeted towards managers trying to digest the scope of the model, new practitioners setting up applications who need to know where to go for further documentation and veteran SELFE users who want a handy reference of tools and assumptions.

The opening image in the guide shows the entire modeling process at a high level. Colored boxes on this diagram represent complete tasks such as setting up the ocean boundary condition. Each of these major components is associated with a sub-process diagram in a matching color later in the guide that breaks the job into smaller pieces. All the diagrams are accompanied by a brief text summary of concepts, tools, resources and documentation. The final pages include some additional reference lists of input and outputs files that users find handy

This is the first version of this guide. If you find yourself referring to material that isn't mentioned here or scribbling information in the margins please let us know. We'd like to improve the usefulness of this guide.

The Bay-Delta SELFE Modeling Team

I Bay-Delta SELFE



I Bay-Delta SELFE

Toolset

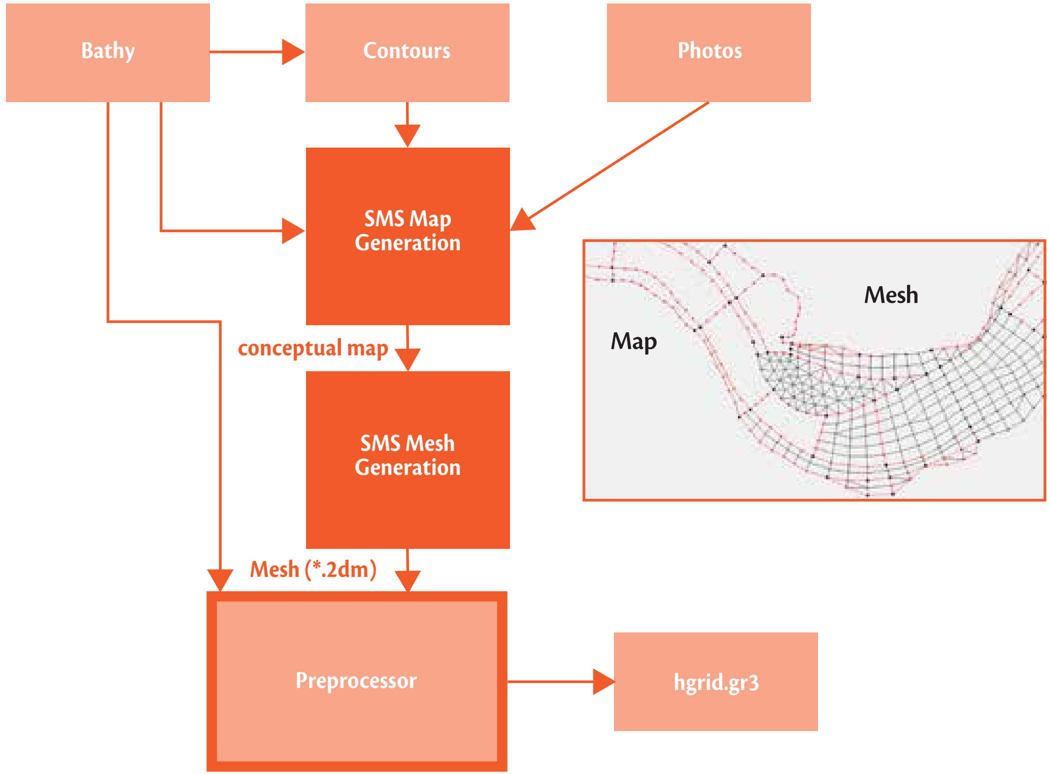
SELFE-W 4.1 (William and Mary version)
SELFE executable utilities
Bay-Delta SELFE Python preprocessing suite
LLNL VisIt visualization
Aquaveo SMS mesh generation
VTide tidal analysis
VTools time series scripting tools for Python
NetCDF

Documentation

VIMS SELFE-W Wiki (http://ccrm.vims.edu/w)
OHSU v3 SELFE documentation
(http://www.stccmop.org/knowledge_transfer/software/selfe/v3manual)
SELFE-W reference documentation (in preparation)
Bay-Delta Package documentation
- VTools html documentation
- Preprocessor html documentation
- Hydraulic Structures for SELFE (pdf)
- VisIt Visualization for SELFE (pdf)
- VTide how-to guide (pdf)

Aquaveo documentation
XMS wiki site (http://www.xmswiki.com/xms/)
NetCDF utilities: http://www.unidata.ucar.edu/software/netcdf/

2 Mesh Generation



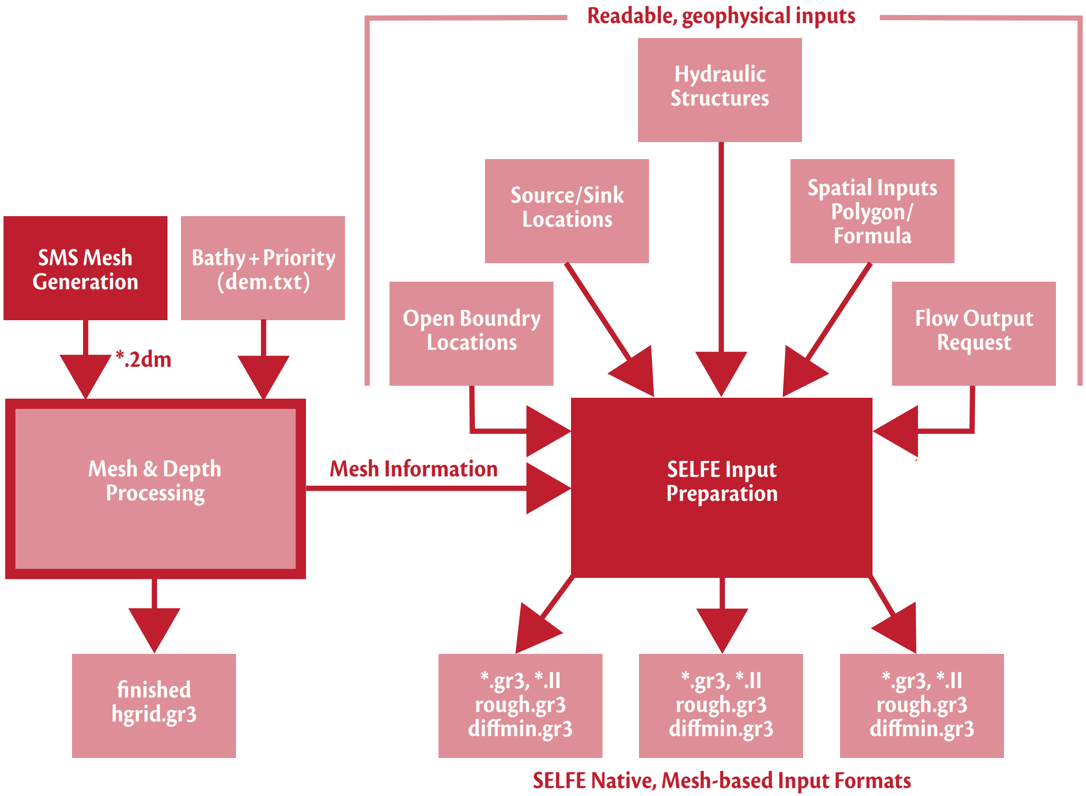
2 Mesh Generation

Concepts
Develop in Aquaveo SMS using a "Conceptual Map"
Wetting and drying considerations for node placement.
Volumetric optimization
Open boundaries must stay wet (artificial deepening)

Tools and Data
Aquaveo SMS Generic (proprietary)
Bay-Delta bathymetry in 2m/10m tiffs
dem.txt prioritizing the DEMs (in package)
contour_smooth.py for curvature flow elevation simplification
Preprocessor to populate and optimize elevations

Detailed Information
XMS wiki and documentation
SELFE Scripting html documentation (in package)
Training and Tutorial session: Mesh Generation

3 Prepocessor



3 Preprocessor

Concepts

Preprocessor converts user-readable inputs to SELFE input
- SELFE native inputs are based on mesh ID locations
- Preprocessor is based on geophysical locations

Utilities for bathymetry smoothing and assignment
Components often run all at once with master input file

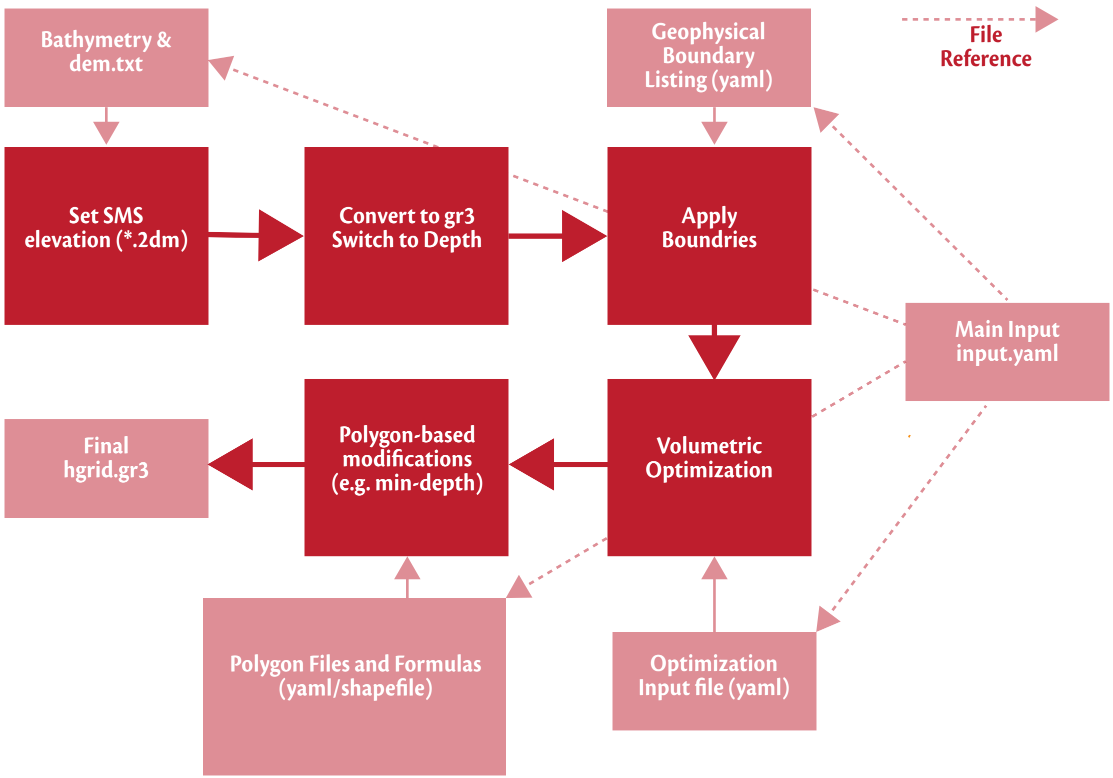
Tools and Data

Input mesh in SMS (2dm) or SELFE (gr3) format
Bathymetry data
Files in yaml format describing locations, parameters
Scripting system for translating inputs to SELFE native

Detailed Information

SELFE Scripting html documentation (in package)
yaml format: www.yaml.org
Training and Tutorial sessions:
- Output
- Mesh Generation
- Hydraulic structures

3A Depth Assignment & Optimization



3A Depth Assignment & Optimization

Concepts

This is a subcomponent of the preprossor
Depths are assigned using a “stack” of prioritized 2m/10m DEMs
Some locations with contorted or undersampled bathymetry (Liberty Island, Sherman Lake) smoothed using a contour-simplifying script

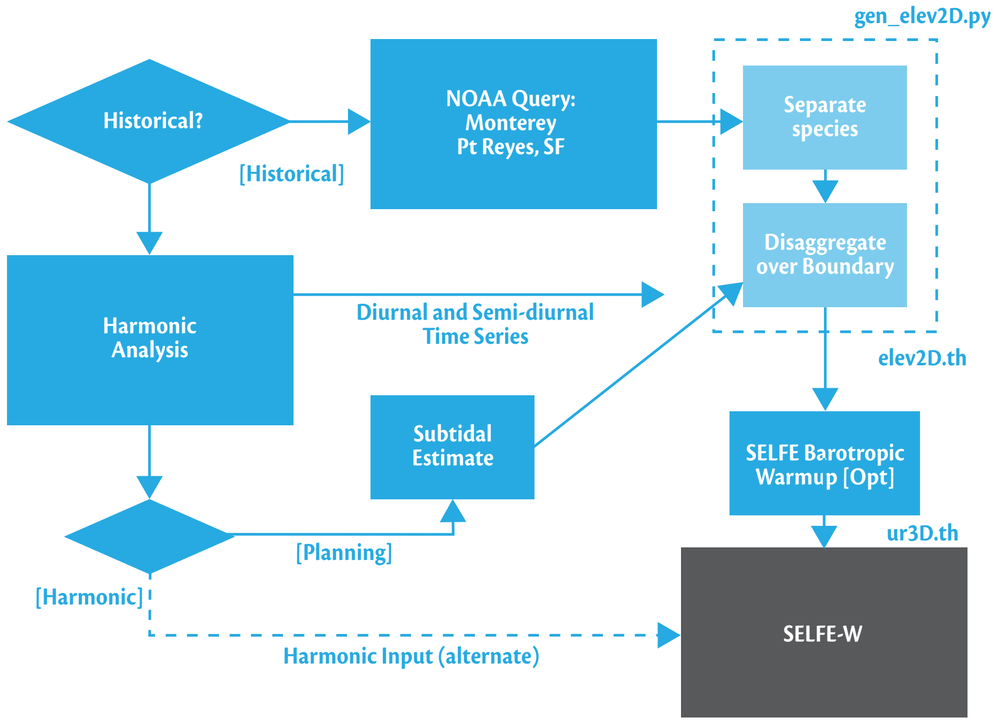
Tools and Data

Input mesh in SMS (2dm) or SELFE (gr3) format
Bay-Delta bathymetry in 2m/10m tiffs
contour_smooth.py script for topographical simplification
Depth optimizing is an automated part of preprocessor

Detailed information

SELFE Scripting documentation and calibration reports

4 Ocean Boundary



4 Ocean Boundary

Concepts and Assumptions

Variation on near coast according to species
- Semi-diurnal amplitude in cross-shore direction
- Semi-diurnal and diurnal phase in alongshore direction
- Subtidal level in cross-shore direction

Underspecification vs overspecification dilemma
- Barotropic-Baroclinic sequence stabilizes boundary velocity

Boundaries can be strict or “nudged” (avoid boundary layer)

Units are NAVD88

Tools and Data

NOAA water levels: Monterey, Pt Reyes

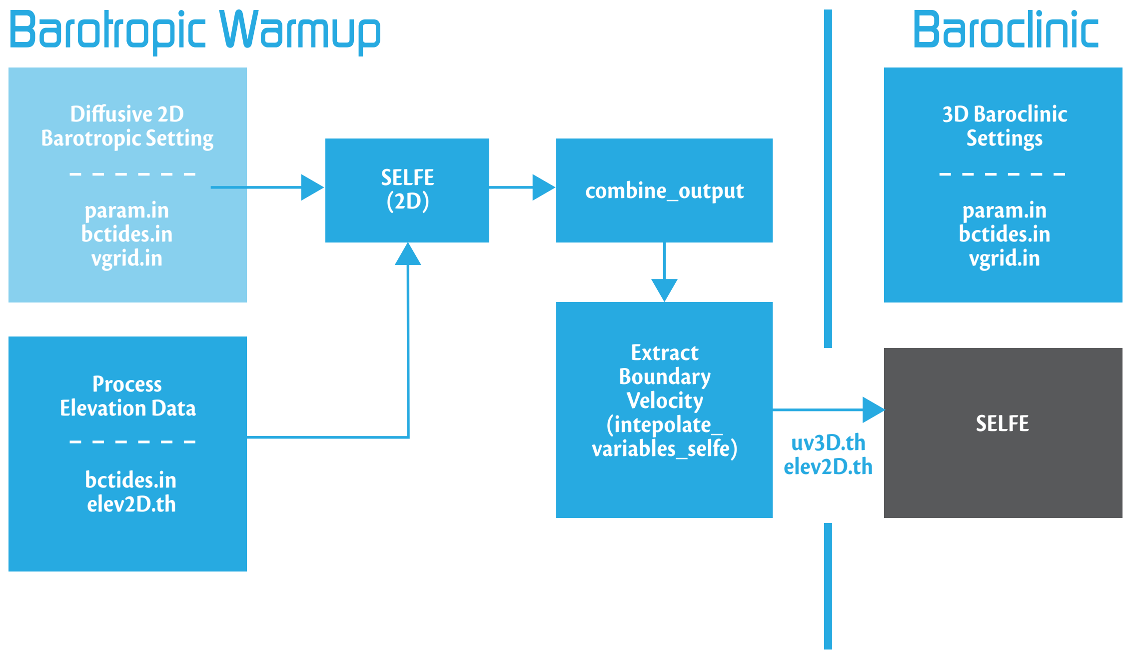
VTide tidal analysis

Package script gen_elev2d.py
- Calls separate_species.py (uses VTools)
- Disaggregates each in along and cross-shore coordinates
- Produces elev2D.th

Detailed information

Training Session: Ocean Boundary

4A Barotropic Warmup



4A Barotropic Warmup

Concepts and Assumptions

Stratified hydrostatic 3D boundaries can be over/underspecified

- Oliger and Sundström (1978) documents hazards of each
- Underspecified means tidal water levels only
- Overspecified means water levels and all velocities

Barotropic Warmup run to produce velocities for baroclinic

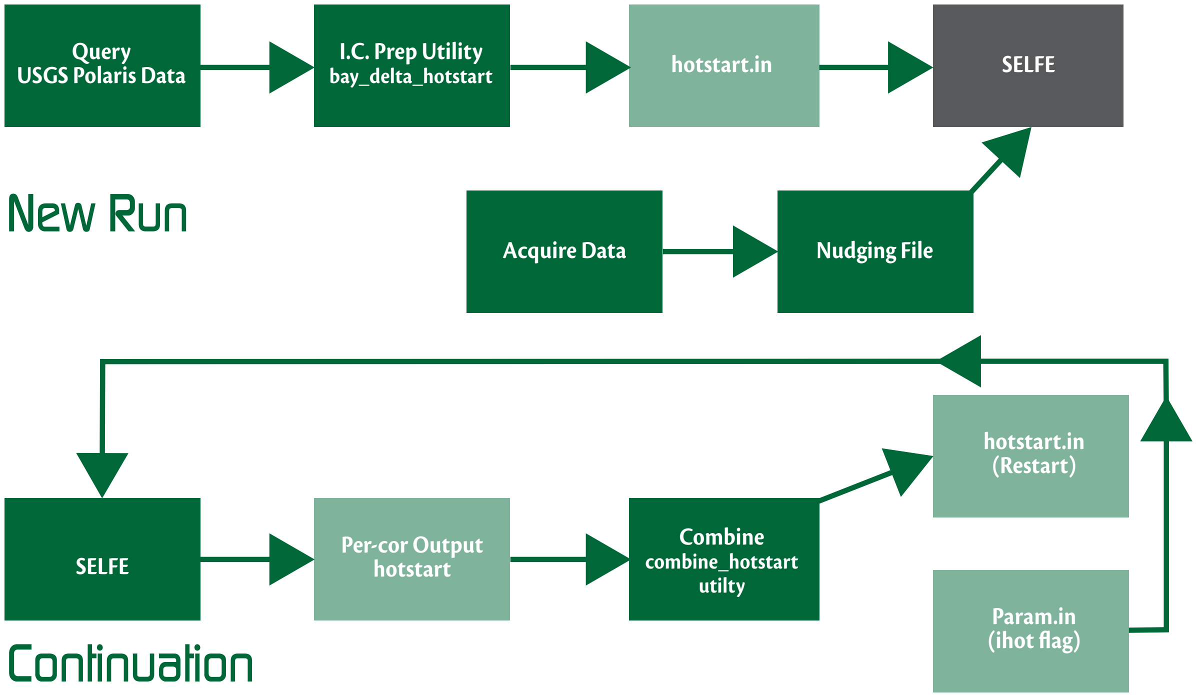
Tools and Data

SELFE 2D diffusive barotropic run
combine_output executable utility
interpolate_variables_selfe executable utility

Detailed information

SELFE barotropic-baroclinic sequence (pdf)

5 Initial Condition



5 Initial Condition

Concepts and Assumptions

SELFE uses hotstart.in to initialize using a full 3D state
Constant I.C. used for elevation (~1m NAVD)
Constant I.C. used for velocity (zero)
Salinity::
- constant for the Ocean (33.5 psu)
- radially extrapolate USGS Polaris data for the Bay
- constant upstream in the Delta (configurable)

Newtonian relaxation ("nudging") reduces spinup in Delta
Simpler text formats (elev.ic, salt.ic) for hypothetical and training

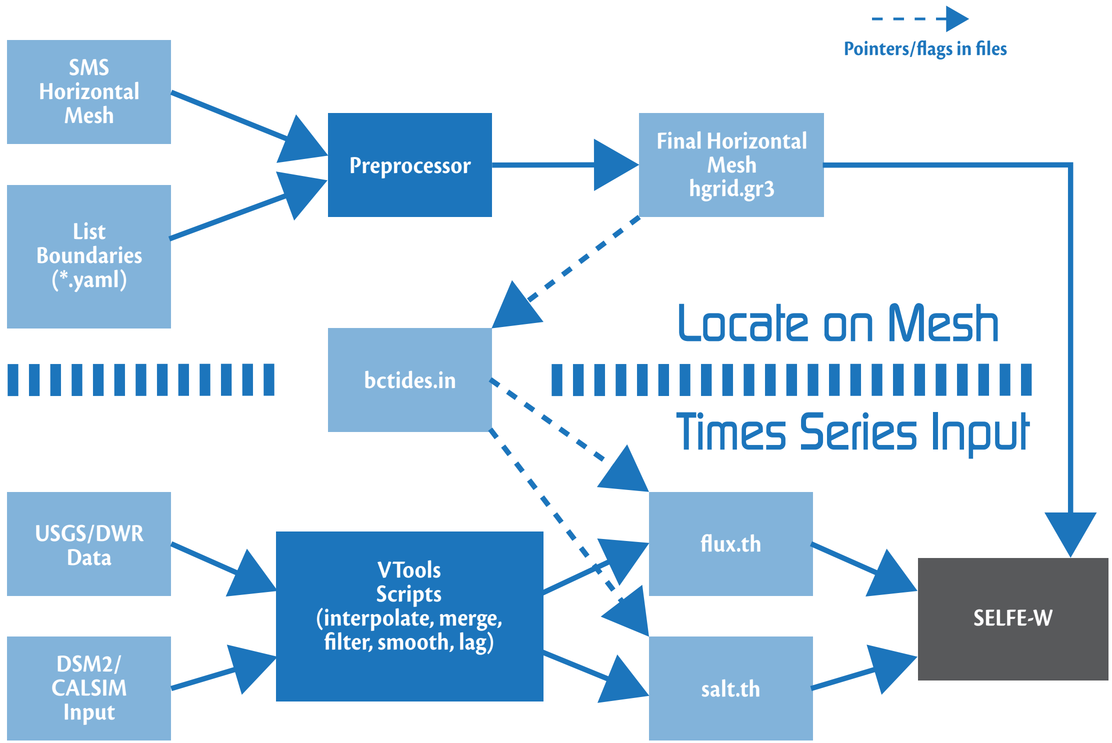
Tools and Data

USGS Polaris cruise data http://sfbay.wr.usgs.gov/access/wqdata/query (simple query)
Executable utility bay_delta_hotstart: for initial data
Execuatble utility combine_hotstart: for restarting

Detailed Information

SELFE reference documentation: hotstart options in param.in.
Training and tutorial session: SELFE Initial Conditions
VIMS SELFE-W Wiki entry on nudginghttp://ccrm.vims.edu/w
VIMS SELFE-W Wiki entry on model initialization

6 Flow Boundaries



6 Flow Boundaries

Concepts and Assumptions

Set data source (const vs time series) flags in bctides.in
Provide actual values in regular time series in *.th format
Boundary data can be "ramped" in on startup,
  avoids IC-BC clash
Sacramento boundary assumptions:
  Sacramento truncated above Fremont Weir, but weir
    not modeled
  Freeport Flow tidally filtered and shifted earlier
    by three hours
  American and Feather Rivers lumped into Sacramento
SELFE will not diffuse unphysical monthly changes in flow
  e.g. Sacramento River in CALSIM should be smoothed.
Daily exports used at most sites
Hourly Banks pumping used in calibration

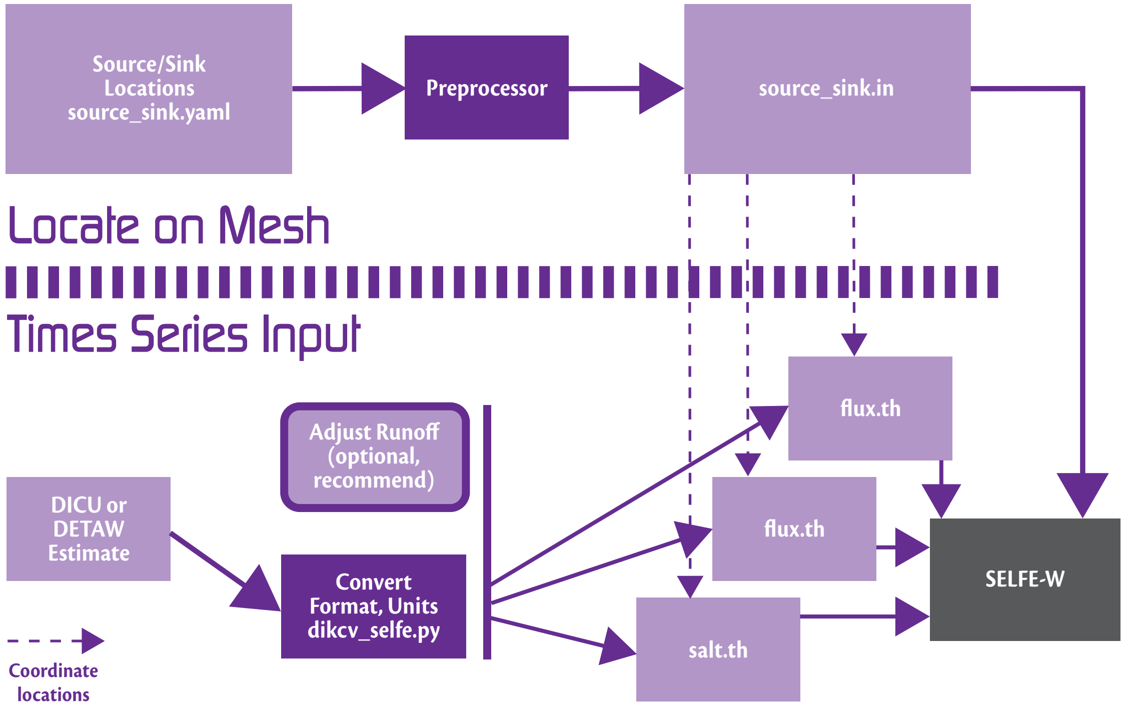
Tools and Data

Data from
  NOAA: ocean boundary
  USGS: Coyote Cr, Napa R., Sacramento R., Vernalis
  DWR: Hourly banks pumping (daily available through CDEC)
  CDEC: Exports from SWP, CVP, CCWD
  DWR Water Data Library: Preferred for DWR NCRO data
  DWR IEP/EMP group: Select IEP and D1641 sites
VTools scripting tools for filtering, shifting and merging

Detailed Information

SELFE and SELFE-W Documentation: bctides.in flags
VIMS SELFE-W Wiki entry on *.th formats
Training Session on Boundaries Conditions

7 Source & Sink



7 Source & Sink

Concepts and Assumptions

DWR DICU model for channel depletions, flows and EC
DETAW or field estimates for updating runoff events

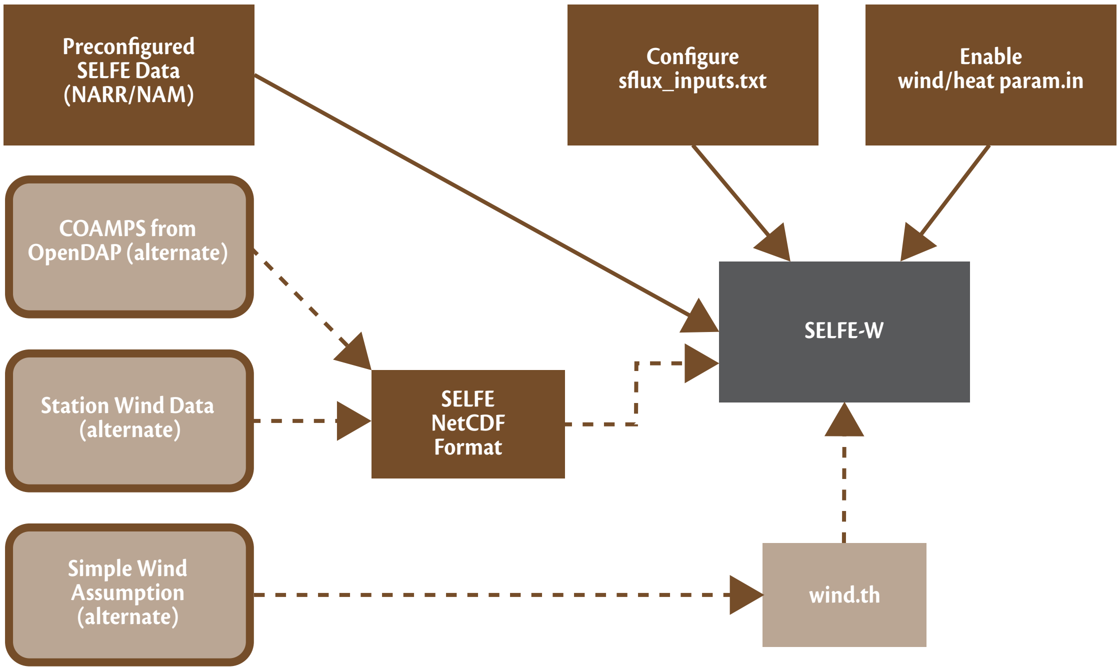
Tools and Data

Package script dicu_selfe.py converts DICU in DSS format.
Package script model_time.py converts elapsed sec in *.th to readable dates

Detailed Information

VIMS SELFE-W Wiki entry on Mass Source and Sink
Training Session (presentation/exercise) on Boundary Conditions

8 Atmospheric Forcing



8 Atmospheric Forcing

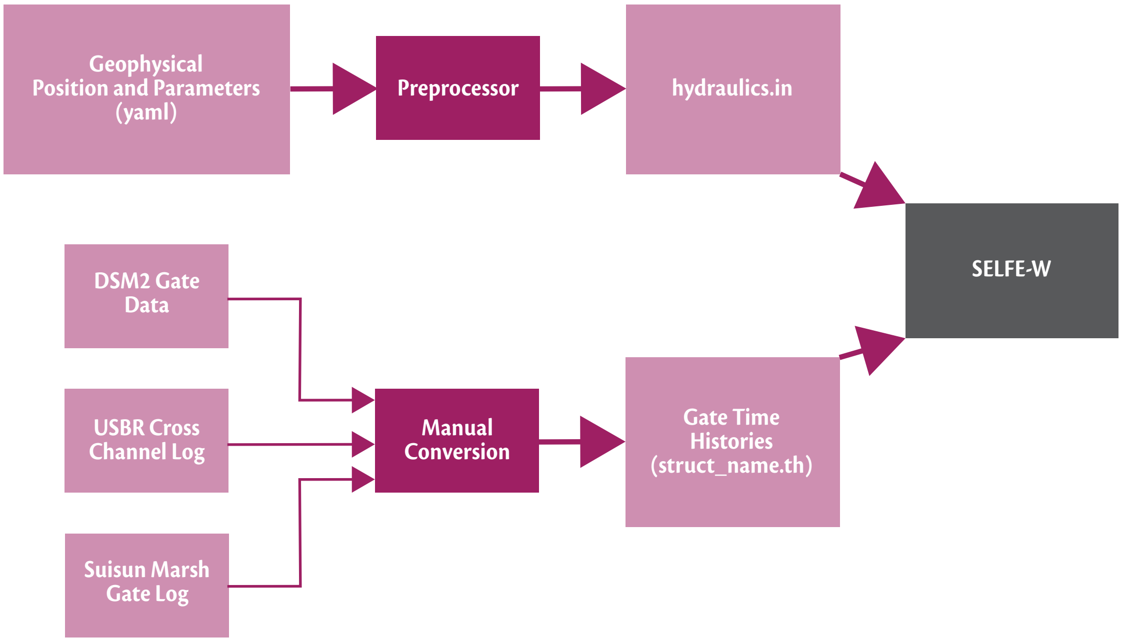
Concepts and Assumptions

SELFE hydrodynamics use wind and atmospheric pressure
SELFE heat module uses precip., radiation flux, humidity
NetCDF (CF) format is used for spatiotemporal inputs
NetCDF 3.0 used by VIMS/OHSU. DWR uses NetCDF 4.0
Calibration used NARR reanalysis data at 32km (see wiki)
Now using CENCOOS (4km) for recent period
(Aug 2013 – 2014)
The direction correction windrot_geo2proj.gr3
is set to zero.
Text wind.th available for practice or hypothetical problems

Tools and Data

Package script cencoos_opendap.py retrieves CENCOOS data
ncdump –h for structure of files (NetCDF distribution)
nccopy for changing NetCDF version (NetCDF distribution)
Preprocessor generates the zeroed-out windrot_geo2proj.gr3

9 Hydraulic Structures



9 Hydraulic Structures

Concepts and Assumptions

Supported structures: weirs, culverts, orifice,
  radial gates, transfers
Replace FEM calculation with subgrid flux estimate
Most major parameters under (irregular) time series control
Can be "uninstalled" in time series
No adaptive operating rule logic
Units are NAVD 88

Tools and Data

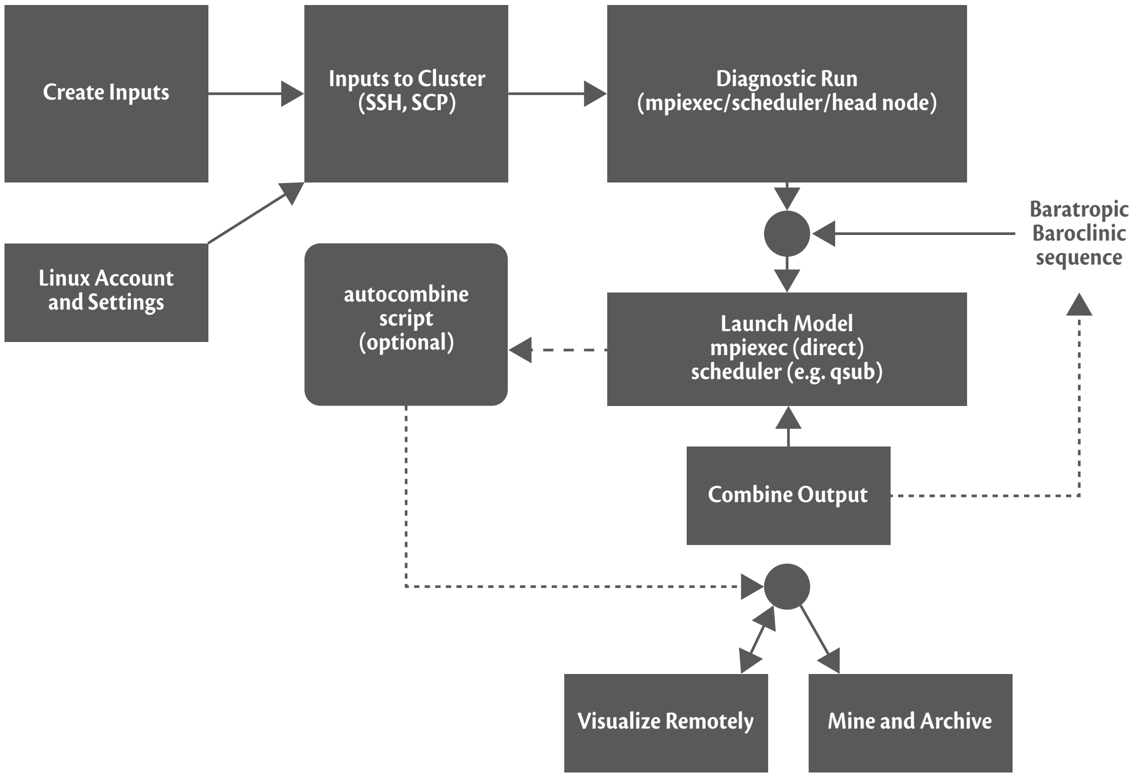
Data sources:
- DSM2 gate inputs, DWR design specifications
- DSM2 Barrier file (DSM2 User Group web site)
  - USGS Cross channel log: http://www.usbr.gov/mp/cvo/vungvari/Ccgates.pdf
- DWR Suisun Marsh Salinity Control History:
  - http://www.water.ca.gov/suisun/dataReports/docs/histsmscgopnew.pdf

Inputs compiled by hand
Package script model_time.py converts elapsed in *.th
  to readable dates

Detailed information

Hydraulic Structures for SELFE (pdf in package)

10 SELFE-W Simulation



10 SELFE-W Simulation

Concepts

Prep work often done in a familiar environment like Windows
Binaries, tutorials and sample simulations also available
    for learning on Windows
Production runs performed on the Linux cluster using
    MPI parallelism
Example resources: DWR clusters, NASA (e.g. Pleiades),
    SDSC (Gordon), Amazon
Equipment investments give good cost/benefit for high use,
    but require IT support
Output can be dense. Postprocessing, reduction, mining in place

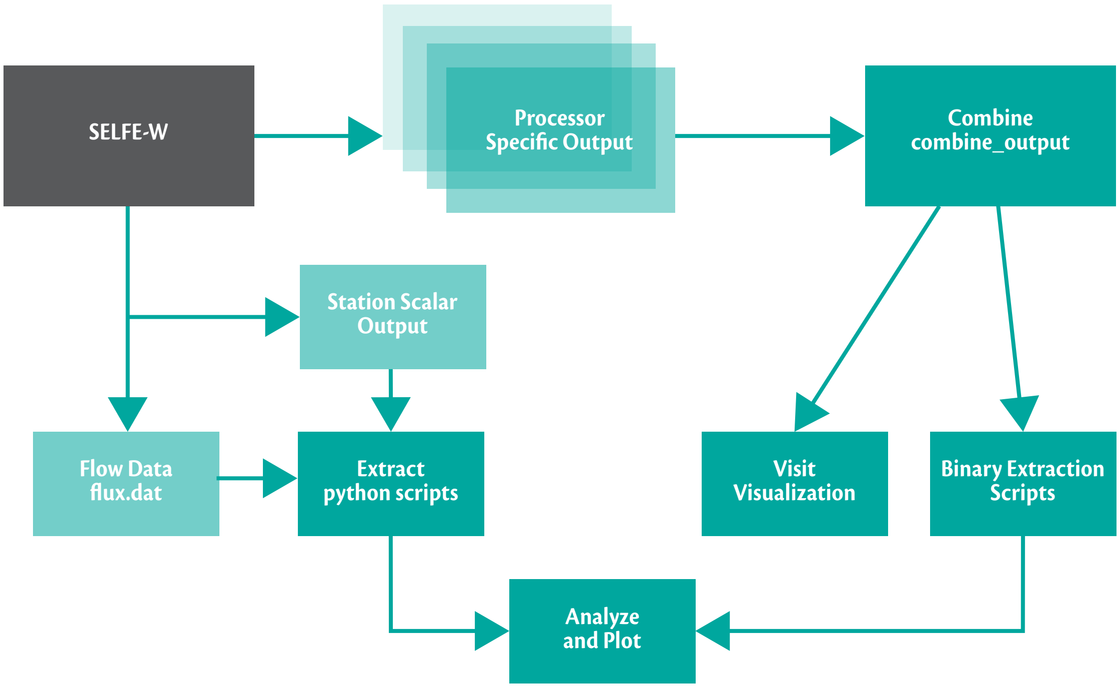
Tools and Data

SELFE model, VIMS (SELFE-W) distribution or patch for OHSU
Bay-Delta SELFE package
Cluster accounts
Compatible compilers (e.g. gcc, Intel, PG)
Prerequisite libraries downloaded (PETSc, ParMETIS, NetCDF)
File transfer tools (SCP, WinSCP)
Linux login and graphic front end (xming)
VisIt plugin (our package) for visualization
    Typically installed on cluster and windows client

Detailed Information

SELFE training materials and documentation
Basic linux tutorials
Documentation and rules for submitting jobs
    Typically a script and one line command
SELFE user group and Delta Modeling Section

Postprocess



II Postprocess

Concepts

SELFE runs well on parallel Linux clusters
  and supercomputers
Preprocessing often done in a familiar environment
Runs performed on the cluster
Some postprocessing can be done in place

Concepts and Assumptions

SELFE outputs full-domain state
  variable selection controlled in param.in
  output interval controlled in param.in
SELFE outputs flux data
  Preprocessor to create request (fluxflags.prop)
  param.in to enable
  Output file in flux.dat
SELFE outputs station (x,y,z) time series:
  request in station.in and param.in
State dumped in processor specific files needs
  to be "combined"
Client-server visualization allows visualization
  of data on cluster
Postprocessing of station time series done using python.

Tools and Data

VisIt uses client server approach
Slicing and extraction done with SELFE utility scripts
SELFE autocombine perl script combines output as-created
Postprocessing tools like metrics.py use matplotlib and vtools
  to compare simulations to each other or to data
OHSU visualization tools

Glossary of files used in SELFE

INPUT:

bctides.in: Boundary condition flags and tidal info
diffmin.gr3, diffmax.gr3: Min. and max. diffusivity (floor and ceiling for turbulence closure)
drag.gr3: Drag coefficients if drag is set directly
fluxflag.prop: Flux output and conservation check file
harm.in: Harmonic analysis input (seldom used)
hgrid.gr3: Horizontal grid and boundary node strings
hgrid.ll: Lat-long version of hgrid.gr3, potentially used for wind interpolation or Coriolis.
hotstart.in: 3D state information for initializing or restarting model
hydraulics.in: Hydraulic structure locations and parameters, enablesstruct_name.th files
interpol.gr3: Interpolation mode: interpol.gr3
krvel.gr3: Kriging flags for interpolation at characteristic feet in ELM
manning.gr3: Friction if Mannings n is used (2D only and currently disabled)
msource.th: Mass flux (temp, salt, tracer) for each source location
param.in: Main input file for parameter, algorithm switches, and output requests.
rough.gr3: Roughness length if drag is derived from roughness and boundary layer.
salt.ic, temp.ic, ts.ic: Initial condition for salt and temperature if not using hotstart.
sflux_air*.nc: Spatiotemporal input for wind speed, pressure, air temp, specific humidity (NetCDF)
sflux_inputs.txt: Atmospheric data timing
sflux_prc*.nc: Spatiotemporal input for precipitation (NetCDF)
sflux_rad*.nc: Spatiotemporal input Long and shortwave radiation forcing (NetCDF)
s_nudge.gr3: Salt nudging horizontal weights
source_sink.in: Locations of sources and sinks

t_nudge.gr3: Salt nudging horizontal weights
tvd.prop: Flag that enables TVD algorithm on a per-element basis
station.in: Station Output request:
vgrid.in: Vertical grid
vsource.th: Flux of water at each source location (see msource.th for concentrations)
vsink.th: Flux of water leaving at each sink location
windrot_geo2proj.gr3: Rotation to align wind input with true north
xlsc.gr3: Surface mixing length

Key Non-Binary Output Files

*(the * indicate processor specific or variable-specific suffixes)*
flux.dat: Fluxes between groups identified in fluxflag.prop
fort.11: diagnostic fatal error message
fort.17: diagnostic number of sub-cycles for upwind/TVD each step
fort.33: diagnostic JCG solver log
global_to_local.prop: Map of global processor mesh numbering to local
local_to_global*: Maps of local processor mesh numbering to global
mirror.out: Main output file including time step and algorithm progress messages
nonfatal_*: Warnings, messages and subcycling messages
staout*: Time series of outputs and variables at (x,y,z) requested in station.in. Files are numbered by variable.

For further information, contact:

Eli Ateljevich, *Project Manager:*
Eli.Ateljevich@water.ca.gov

Tara Smith, *Delta Modeling Section Chief:*
Tara.Smith@water.ca.gov

Departmente of Water Resources
Delta Modeling Section
1416 Ninth St
Sacramento, CA 95814